UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/821,687	04/09/2004	F. Soner Terek	MSFT-2955/307064.01	1355

41505 7590 11/24/2008		EXAMINER
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)		COLAN, GIOVANNA B
CIRA CENTRE, 12TH FLOOR		
2929 ARCH STREET	ART UNIT	PAPER NUMBER
PHILADELPHIA, PA 19104-2891	2162	

MAIL DATE	DELIVERY MODE
11/24/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/821,687	TEREK ET AL.
	Examiner	Art Unit	
	GIOVANNA COLAN	2162	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>22 September 2008</u>.

2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>38, 40 – 49, 54 – 55, and 57 – 58</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) <u>38, 40 – 49, 54 – 55, and 57 – 58</u> is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All b) ☐ Some * c) ☐ None of:

       1. ☐ Certified copies of the priority documents have been received.

       2. ☐ Certified copies of the priority documents have been received in Application No. _____.

       3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date <u>09/22/2008</u>.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1.   This action is issued in response to applicant filed request for continued

examination (RCE) on 09/22/2008.

2.   Claims 38, 40, 45, 46, 49, 54, 55, and 58 have been amended. No claims were

added. Claims 1 – 37, 39, 50 – 53, and 56 were canceled.

3.   Claims 38, 40 – 49, 54 – 55, and 57 – 58 are pending in this application.

4.   Applicant's arguments with respect to amended claims 38, 40, 45, 46, 49, 54, 55,

and 58 have been considered but are moot in view of the new ground(s) of rejection.

Continued Examination Under 37 CFR 1.114

5.   A request for continued examination under 37 CFR 1.114, including the fee set

forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this

application is eligible for continued examination under 37 CFR 1.114, and the fee set

forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action

has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on

10/11/2006 has been entered.

Specification

6.   The specification is objected to as failing to provide proper antecedent basis for

the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction

of the following is required: the term "computer readable medium" recited in claims 38,

49, 55, and 58 lacks of antecedent basis in the specification.

Claim Rejections - 35 USC § 101

7.   35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of
> matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the
> conditions and requirements of this title.

8.   Claims 38, 40 – 49, 54 – 55, and 57 – 58 are rejected under 35 U.S.C. 101

because the claimed invention is directed to non-statutory subject matter.

        Claims 38, 49, 55, and 58 recite a "computer-readable medium ... for retrieval by

computer hardware". However, applicant has failed to provide antecedent basis for the

claim terminology "computer readable medium". Examiner makes note that; "for

retrieval by computer hardware..." does not necessarily mean

including/comprising/consisting of a computer hardware. Also, the specification

discloses that the claimed invention is directed to software only (Also, see [0187], lines

7 – 11, "implemented in connection with hardware or software or, where appropriate,

with a combination of both", Specification of the disclosure).

        Therefore, the claims become nothing more than sets of software instructions

which are "software per se".

Claim Rejections - 35 USC § 103

9.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

        This application currently names joint inventors. In considering patentability of

the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of

the various claims was commonly owned at the time any inventions covered therein

were made absent any evidence to the contrary. Applicant is advised of the obligation

under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was

not commonly owned at the time a later invention was made in order for the examiner to

consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g)

prior art under 35 U.S.C. 103(a).

10.   Claims 38 and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable

over Bennion (U.S. Patent No. 5,634,123).

        Regarding claim 38, Bennion teaches a computer readable medium bearing a

computer readable representation ... comprising:

        a binary fragment (Fig. 2, item 201, Col. 7, lines 7 – 10, Bennion) associated with

said object, said binary fragment comprising a binary fragment header and a binary

fragment payload (col. 1, lines 56-60, "The data structure defined by the present

invention includes two types of records: data-containing records and container records.

Data-containing records contain data, while container records contain other records. A

code point found at the beginning of each record specifies its type", wherein a code

point is equivalent to a header and one of the types is data-containing records, in which

case the header is a binary fragment header; Figure 2, element 205, wherein the Data

portion is equivalent to the payload, Bennion[1]), wherein the binary fragment payload

includes a plurality of primitive data members of said object (col. 3, lines 19-22, wherein

the "Company" (Figure 6, item 603, Bennion) corresponds to the object claimed,

Bennion) and also since "COMPANY DATA 601" is a data-containing record that

contains data only (not other records), Bennion), and wherein the binary fragment

payload is devoid of any non-primitive members of said object (Fig. 6, item 601, Col. 3,

lines 19 – 27, "...Container records contain other records, while data-containing records

contain data. No record is both container record and a data-containing record..."};

wherein the Examiner interprets that since item 601 "COMPANY DATA", Fig. 6, does

not contain other records, as opposed to "REGION 602", then it is devoid of any non-

primitive member as claimed, Bennion[2]); and

at least one additional fragment comprising at least one non-primitive member of

said object (See at least col. 1, lines 56-60, "The data structure defined by the present

---

[1] Wherein the "Company" (Figure 6, item 603, Bennion) corresponds to the object claimed.
[2] Examiner makes note that the specification provides examples of "primitive members", (specification of
the disclosure; [0006], "string", "integer") also as defined by the dictionary ("Academic Press Dictionary of
Science and Technology from Elsevier Science & Technology, Copyright 1992, 1996 by Academic
Press") Primitive is a fundamental unit that cannot be divided. Therefore, data (in a data-containing
record which does NOT contain other records) is of a primitive data type. See also, figure 3, Bennion, first
block of data starting from the left on "151-200", the payload "DATA" includes "123 Avenue "E" " which
are string and integers. Examiner has interpreted the claims in light of the specification. See *In re Van
Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type"; Figure 3, and col. 6, lines 5-18, wherein "a sample string of data bytes representing a series of hierarchically-organized records" illustrates a fragment comprising non-primitive members of the object, Bennion).

Bennion does not explicitly teach said plurality of primitive data members in the binary fragment payload represent "all" the primitive data members associated with said object. However, since each record within the COMPANY object may be classified as either a container record or a data-containing record (i.e. a primitive data member) (col. 3, lines 19-22, wherein the "Company" (Figure 6, item 603, Bennion) corresponds to the object claimed, Bennion) and also since "COMPANY DATA 601" is a data-containing record that contains data only (not other records) and also since it "describes the company as a whole", then it is suggested that data-containing record "COMPANY DATA 601" must represent all the primitive data members associated with the object "COMPANY" (Col. 3, lines 5 – 8, Bennion). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include all of the primitive data members of the object in a binary fragment payload as taught by Bennion to provide a complete COMPANY object record as shown in the example given in col. 3 lines 19-31. One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

Bennion also discloses:

wherein said plurality primitive data members are in a storage engine record

format (col. 3, lines 19-24, "Thus, a hierarchical structure emerges. Each individual

record contained within COMPANY record 600 may be classified as either a container

record or a data-containing record. Container records contain other records, while data-

containing records contain data. No record is both a container record and a data-

containing record"; col. 4, lines 40-43, "By definition, a data-containing record has a

"Length" equal to the actual length of the record itself, since a data-containing record

cannot contain another record", wherein a data-containing record is a primitive data

member, Bennion),

wherein the binary fragment header comprises a type field (col. 1, lines 56-60,

"The data structure defined by the present invention includes two types of records: data-

containing records and container records. Data-containing records contain data, while

container records contain other records. A code point found at the beginning of each

record specifies its type.", wherein the data-containing record type is equivalent to a

binary fragment type) and a length field (Figure 2.; col. 1, lines 60-62, "A length field

facilitates variable data lengths for data-containing records", Bennion), and

wherein the type field indicates that the binary fragment payload is devoid of any

non-primitive data members (Fig. 6, item 601, Col. 3, lines 19 – 27, "…Container

records contain other records, while data-containing records contain data. No record is

both container record and a data-containing record…"; wherein the Examiner interprets

that since item 601 "COMPANY DATA", Fig. 6, does not contain other records, as

opposed to "REGION 602", then it is devoid of any non-primitive member as claimed,

Bennion[3]).

Claim 44 is rejected for the reasons set forth hereinabove for claim 38 and

furthermore Bennion discloses a medium comprising a terminator fragment that marks

the end of the object, said terminator fragment comprising a terminator type field

indicating the terminator fragment is a terminator fragment (col. 6, lines 40-42,

Bennion).

12.   Claims 40-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Bennion (U.S. Patent No. 5,634,123), in view of Krishnaprasad et al, "Krishnaprasad"

(U.S. Publication No. 2004/0220946), and further in view of Sarkar (U.S. Patent No.

6,012,067).

Claim 40 is rejected for the reasons set forth hereinabove for claim 38. However,

Bennion does not explicitly disclose a fragment comprising Large Object (LOB)

fragment.

Krishnaprasad discloses a computer readable medium comprising:

---

[3] Examiner makes note that the specification provides examples of "primitive members", (specification of the disclosure; [0006], "string", "integer") also as defined by the dictionary ("Academic Press Dictionary of Science and Technology from Elsevier Science & Technology, Copyright 1992, 1996 by Academic Press") Primitive is a fundamental unit that cannot be divided. Therefore, data (in a data-containing record which does NOT contain other records) is of a primitive data type. See also, figure 3, Bennion, first block of data starting from the left on "151-200", the payload "DATA" includes "123 Avenue "E" " which

at least one Large Object (LOB) fragment comprising a LOB fragment header (page 5, section [0062], "Message 300 includes four fields: a length field 302; a version field 304; a flag field 306; and a payload field 310. The combination of fields 302, 304, 306, 310 composes a serialized image of XML data, according to the illustrated embodiment", wherein the fields except the payload field are considered to be the header portion) and a LOB fragment payload (page 6, section [0066], "Payload field 310 includes serialized XML data for a particular XML construct"),

wherein the LOB header comprises a LOB type field, wherein the LOB type field indicates the LOB fragment is a LOB fragment (page 6, section [0066], "If the flag field 306 indicates that the type is LOB, then a locator for the LOB appears in the payload, such as the octal string "0.times.000030303030303- 0 . . . "), and a LOB length field, wherein the LOB length field indicates a length of the LOB fragment payload (page 5, section [0063], "Length field 302 includes data that indicates the length of the serialized image. Any method known in the art for indicating length may be used."; Page 6, TABLE 2).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate a LOB fragment structure as disclosed by Krishnaprasad into the computer readable representation as disclosed in Bennion so that an XML element may be transferred between processes 132a and 132b, which both have access to LOB 144b, by passing the LOB locator (page 4, section [0047]).

---

are string and integers. Examiner has interpreted the claims in light of the specification. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

However, the combination of Bennion in view of Krishnaprasad does not explicitly disclose a value type field that indicates whether the LOB fragment payload comprises an inline LOB or a pointer to a LOB location.

Sarkar teaches a value type field, wherein the value type field indicates whether the LOB fragment payload comprises an inline LOB or a pointer to a LOB location (col. 4, lines 25-33, "Internal LOB columns contain LOB locators that can refer to out-of-line or inline LOB values. Selecting a LOB column value returns the LOB locator and not the entire LOB value. Different operations in the form of packages and functions are performed through these locators. Multiple LOB data type columns can be defined in a table and all possible SQL operations are possible over such tables and attributes. LOB locator can be stored in the table column, either with or without the actual LOB value").

It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate a value type field, as disclosed by Sarkar, into the computer readable representation as disclosed in the combination of Bennion in view of Krishnaprasad so that when a LOB column value is selected, the LOB locator is first returned instead of the entire LOB value (col. 4, lines 27-28). One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

Claim 41 is rejected for the reasons set forth hereinabove for claim 40 and furthermore the combination of Bennion in view of Krishnaprasad and further in view of Sarkar discloses a medium wherein the LOB fragment payload comprises a LOB (col. 4, lines 25-33, Sarkar).

Claim 42 is rejected for the reasons set forth hereinabove for claim 40 and furthermore the combination of Bennion in view of Krishnaprasad and further in view of Sarkar discloses a medium wherein the LOB fragment payload comprises a pointer to a LOB location (col. 4, lines 25-33, Sarkar).

Claim 43 is rejected for the reasons set forth hereinabove for claim 40 and furthermore the combination of Bennion in view of Krishnaprasad and further in view of Sarkar discloses a medium wherein the value type field indicates whether the LOB fragment payload comprises an inline LOB, a pointer to a LOB location, or a cell reference (col. 4, lines 25-33; col. 5, lines 18-47, Sarkar).

13.   Claims 45-48 and 55-57 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bennion (U.S. Patent No. 5,634,123), in view of Stickler (US Patent No. 6,904,454 B2).

Claim 45 is rejected for the reasons set forth hereinabove for claim 38 and furthermore Bennion discloses a medium wherein said at least one additional fragment comprises:

a collection start fragment comprising a collection start header (col. 1, lines 56-60, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. **A code point found at the beginning of each record specifies its type**", wherein the fragment indicating the code element corresponds to the collection start fragment as claimed; also Examiner makes note that Bennion's code point is at the beginning of each record; therefore, the code is a collection start fragment as claimed; Bennion[4]),

wherein the collection start header comprises a collection start type field, wherein the collection start type field indicates the collection start fragment is a collection start fragment (Col. 1, lines 56 - 63, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. **A code point found at the beginning of each record specifies its type**. A length field facilitates variable data lengths for data-containing records, as well as implicit definition of a hierarchical structure among records", wherein container records are equivalent to "collection". "… a hierarchical structure among records…" indicates the structure of a collection type record; col. 5, lines 18-20, "**One bit is used to indicate that this is a**

---

[4] Also see, Col. 6, lines 12 – 16, Bennion, "In the code points shown, the first byte is CO for container records ...". As also shown in Figure 3, first row 1 – 50, Bennion, the "CP" (code point) is always at the start of the blocks of data. For example, first block CP=CONT with LEN=302, second block CP=CONT with LEN=98, third block CP...etc.

container record (as opposed to a data-containing record)") indicate that this is a

container record (as opposed to a data-containing record)", Bennion[5]), and

a plurality of collection element fragments (See at least col. 1, lines 56 – 63;

Figure 3, and col. 6, lines 5-18).

However, Bennion does not explicitly disclose a bit field, wherein the bit field

indicates whether an order exists among a plurality of collection element fragments.

Stickler, on the other hand, discloses a bit field, wherein the bit field indicates

whether an order exists among a plurality of collection element fragments (Col. 1 and

17, lines 56 – 58 and 45 – 47; respectively, "at least one entity includes metadata that

identifies a sequential relationship between one or more entities within the scope of said

one entity, each of said entities including metadata defining a position within said

sequential relationship", wherein the one entity that identifies a sequential relationship

(i.e. order of entities) corresponds to the bit field that indicates whether an order exists

among collection element fragments as claimed, Stickler). It would have been obvious

to one having ordinary skill in the art at the time the invention was made to modify the

teachings of Bennion by incorporating a bit field indicating whether an order exists as

disclosed by Stickler among a plurality of collection element fragments, to overcome a

difficult situation present in known tree-based versioning models namely their inability to

explicitly define relationships between different releases (col. 2, lines 1-5). Furthermore,

---

[5] Also see, Col. 6, lines 12 – 16, Bennion, "In the code points shown, the first byte is CO for container records ...". As also shown in Figure 3, first row 1 – 50, Bennion, the "CP" (code point) is always at the start of the blocks of data. For example, first block CP=CONT with LEN=302, second block CP=CONT with LEN=98, third block CP...etc.

MARS 25 also provides encoding properties defining special qualities relating to the format, structure or general serialization of data streams (col. 10, lines 16-18). One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

Claim 46 is rejected for the reasons set forth hereinabove for claim 45 and furthermore the combination of Bennion in view of Stickler discloses a medium comprising:

at least one collection element fragment comprising a collection element header and collection element payload (col. 1, lines 56-60, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type"; col. 5, lines 18-20, "One bit is used to indicate that this is a container record (as opposed to a data-containing record)", wherein a code point is equivalent to a header and one of the types is container records, in which case the header is a collection element header because container records contain other records, which is equivalent to a "collection" element, and the data itself is the payload, Bennion),

wherein the collection element header comprises a collection element type field and a collection element length field, wherein the collection element type field indicates the collection element fragment is a collection element fragment (col. 5, lines 18-20,

"One bit is used to indicate that this is a container record (as opposed to a data-

containing record)", Bennion), and

      wherein the collection element length field indicates the a length of the collection

element payload (col. 5, lines 23-25, "For container records, the length field 203

specifies the total length of all records that the current record contains (as described

above)", Bennion).

      Claim 47 is rejected for the reasons set forth hereinabove for claim 46, and

furthermore the combination of Bennion in view of Stickler discloses a medium wherein

the collection element payload comprises a data member in a collection of data

members corresponding to said collection start fragment (col. 3, lines 32-43, Bennion).

      Claim 48 is rejected for the reasons set forth hereinabove for claim 46, and

furthermore the combination of Bennion in view of Stickler discloses a medium wherein

      the collection element header further comprises a collection element locator field

that provides a unique location of a data member in a collection of data members (col.

4, lines 24-31, Bennion).

      Claim 55 is rejected on grounds corresponding to the reasons given above for

claims 45 and 46.

Claims 56, 57 are rejected on grounds corresponding to the reasons given above

for claims 47 and 48.

14.   Claims 49 and 54 are rejected under 35 U.S.C. 103(a) as being unpatentable

over Bennion (U.S. Patent No. 5,634,123), in view of Krishnaprasad et al,

"Krishnaprasad" (U.S. Publication No. 2004/0220946), in view of Sarkar (U.S. Patent

No. 6,012,067), and further in view of Stickler (US Patent No. 6,904,454 B2).

With respect to claim 49, Bennion teaches a computer readable medium bearing

a computer readable representation … comprising:

a collection start fragment comprising a collection start header (col. 1, lines 56-

60, "The data structure defined by the present invention includes two types of records:

data-containing records and container records. Data-containing records contain data,

while container records contain other records. A code point found at the beginning of

each record specifies its type", wherein container records are equivalent to "collection".

A code point is equivalent to the start header, Bennion),

wherein the collection start header comprises a collection start type field, wherein

the collection start type field indicates the collection start fragment is a collection start

fragment (col. 1, lines 56 - 63, "The data structure defined by the present invention

includes two types of records: data-containing records and container records. Data-

containing records contain data, while container records contain other records. A code

point found at the beginning of each record specifies its type. A length field facilitates

variable data lengths for data-containing records, as well as implicit definition of a

hierarchical structure among records", wherein container records are equivalent to

"collection". A code point is equivalent to the start header. "a hierarchical structure

among records" indicates the structure of a collection type record; col. 5, lines 18-20,

"One bit is used to indicate that this is a container record (as opposed to a data-

containing record)", Bennion), and

     the plurality of collection element fragments (See at least col. 1, lines 56 – 63;

Figure 3, and col. 6, lines 5-18, Bennion) associated with said collection start fragment

(Fig. 3, Col. 6, lines 5 – 15, "…the first byte is CO for container records…", Bennion),

wherein each of the plurality of collection element fragments comprises a collection

element header and a collection element payload, and wherein each collection payload

comprises a data member of a collection element data type (Col. 5 and 6, lines 5 – 13

and 9 – 15; respectively, Bennion), said collection element data type comprising data of

same type as every collection element associated with said collection start fragment

(Col. 3 and 6, lines 19 – 24 and 9 – 16; "No record is both a container record and a

data-containing record…" and "…and 'DATA' is data field. In the code points shown, the

first byte is CO for container records…"; respectively, wherein the first byte CO indicates

that the data is of same type, such as, container record, Bennion).

     However, Bennion does not explicitly disclose a fragment comprising Large

Object (LOB) fragment and nor does Bennion explicitly disclose a bit field, wherein the

bit field indicates whether an order exists among a plurality of collection element
fragments.

Krishnaprasad discloses a computer readable medium bearing ... representation
... comprising:

at least one Large Object (LOB) fragment comprising a LOB fragment header
(page 5, section [0062], "Message 300 includes four fields: a length field 302; a version
field 304; a flag field 306; and a payload field 310. The combination of fields 302, 304,
306, 310 composes a serialized image of XML data, according to the illustrated
embodiment", wherein the fields except the payload field are considered to be the
header portion) and a LOB fragment payload (page 6, section [0066], "Payload field 310
includes serialized XML data for a particular XML construct", Krishnaprasad),

wherein the LOB header comprises a LOB type field, wherein the LOB type field
indicates the LOB fragment is a LOB fragment (page 6, section [0066], "If the flag field
306 indicates that the type is LOB, then a locator for the LOB appears in the payload,
such as the octal string "0.times.000030303030303- 0 . . . ", Krishnaprasad),

and a LOB length field, wherein the LOB length field indicates a length of the
LOB fragment payload (page 5, section [0063], "Length field 302 includes data that
indicates the length of the serialized image. Any method known in the art for indicating
length may be used."; Page 6, TABLE 2, Krishnaprasad).

It would have been obvious to one having ordinary skill in the art at the time the
invention was made to incorporate a LOB fragment structure as disclosed by
Krishnaprasad into the computer readable representation as disclosed in Bennion so

that an XML element may be transferred between processes 132a and 132b, which

both have access to LOB 144b, by passing the LOB locator (page 4, section [0047]).

One of ordinary skill in the art would be motivated to make the aforementioned

combination with reasonable expectation of success.

However, the combination of Bennion in view of Krishnaprasad does not explicitly

disclose a value type field that indicates whether the LOB fragment payload comprises

an inline LOB or a pointer to a LOB location, nor a bit field, wherein the bit field indicates

whether an order exists among a plurality of collection element fragments. Sarkar

teaches a value type field, wherein the value type field indicates whether the LOB

fragment payload comprises an inline LOB or a pointer to a LOB location (col. 4, lines

25-33, "Internal LOB columns contain LOB locators that can refer to out-of-line or inline

LOB values. Selecting a LOB column value returns the LOB locator and not the entire

LOB value. Different operations in the form of packages and functions are performed

through these locators. Multiple LOB data type columns can be defined in a table and all

possible SQL operations are possible over such tables and attributes. LOB locator can

be stored in the table column, either with or without the actual LOB value", Sarkar). It

would have been obvious to one having ordinary skill in the art at the time the invention

was made to incorporate a value type field, as disclosed by Sarkar, into the computer

readable representation as disclosed in the combination of Bennion view of

Krishnaprasad so that when a LOB column value is selected, the LOB locator is first

returned instead of the entire LOB value (col. 4, lines 27-28, Sarkar). One of ordinary

skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

However, the combination of Bennion, in view of Krishnaprasad, and further in view of Sarkar does not explicitly disclose a bit field, wherein the bit field indicates a property of a collection in one of a plurality of collection element fragments, wherein the bit field further indicates whether an order exists among the plurality of collection element fragments. Stickler discloses a bit field, wherein the bit field indicates a property of a collection in one of a plurality of collection element fragments (Col. 7, lines 40 – 49, and Col. 17, lines 45 – 47, Stickler), wherein the bit field further indicates whether an order exists among the plurality of collection element fragments (col. 1, lines 56-58, "at least one entity includes metadata that identifies a sequential relationship between one or more entities within the scope of said one entity, each of said entities including metadata defining a position within said sequential relationship", wherein the one entity that identifies a sequential relationship (i.e. order of entities) is analogous to the bit field that indicates whether an order exists among collection element fragments as claimed, Stickler). It would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate a bit field indicating whether an order exists as disclosed by Stickler among a plurality of collection element fragments, as disclosed in Bennion, to overcome a difficult situation present in known tree-based versioning models namely their inability to explicitly define relationships between different releases (col. 2, lines 1-5, Stickler). Furthermore, MARS 25 also provides encoding properties defining special qualities relating to the format, structure or general

serialization of data streams (col. 10, lines 16-18). One of ordinary skill in the art would be motivated to make the aforementioned combination with reasonable expectation of success.

Claim 54 is rejected for the reasons set forth hereinabove for claim 53 and furthermore Bennion, in view of Krishnaprasad, in view of Sarkar, and further in view of Stickler discloses a medium comprising:

Wherein the collection element header comprises a collection element type field and collection element length field (col. 1, lines 56-60, "The data structure defined by the present invention includes two types of records: data-containing records and container records. Data-containing records contain data, while container records contain other records. A code point found at the beginning of each record specifies its type"; col. 5, lines 18-20, "One bit is used to indicate that this is a container record (as opposed to a data-containing record)", wherein a code point is equivalent to a header and one of the types is container records, in which case the header is a collection element header because container records contain other records, which is equivalent to a "collection" element, and the data itself is the payload, Bennion),

wherein the collection element header comprises a collection element type field and a collection element length field, wherein the collection element type field indicates the collection element fragment is a collection element fragment; (col. 5, lines 18-20, "One bit is used to indicate that this is a container record (as opposed to a data-containing record)", Bennion), and

wherein the collection element length field indicates a length of the collection

element payload (col. 5, lines 23-25, "For container records, the length field 203

specifies the total length of all records that the current record contains (as described

above)", Bennion).

15.   Claim 58 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bennion

(U.S. Patent No. 5,634,123) in view of Roy et al. ("Roy" hereinafter) (US Patent No.

6,631,130).

        Regarding claim 58, Bennion teaches a computer readable medium bearing a

computer readable representation ... comprising:

        a binary fragment (Fig. 2, item 201, Col. 7, lines 7 – 10, Bennion) associated with

said object, said binary fragment comprising a binary fragment header and a binary

fragment payload (col. 1, lines 56-60, "The data structure defined by the present

invention includes two types of records: data-containing records and container records.

Data-containing records contain data, while container records contain other records. A

code point found at the beginning of each record specifies its type", wherein a code

point is equivalent to a header and one of the types is data-containing records, in which

case the header is a binary fragment header; Figure 2, element 205, wherein the Data

portion is equivalent to the payload, Bennion[6]), wherein the binary fragment payload

includes a plurality of primitive data members of said object (col. 3, lines 19-22, wherein

the "Company" (Figure 6, item 603, Bennion) corresponds to the object claimed,

---

[6] Wherein the "Company" (Figure 6, item 603, Bennion) corresponds to the object claimed.

Bennion) and also since "COMPANY DATA 601" is a data-containing record that

contains data only (not other records), Bennion), and is devoid of any non-primitive

members of said object (Fig. 6, item 601, Col. 3, lines 19 – 27, "…Container records

contain other records, while data-containing records contain data. No record is both

container record and a data-containing record…"; wherein the Examiner interprets that

since item 601 "COMPANY DATA", Fig. 6, does not contain other records, as opposed

to "REGION 602", then it is devoid of any non-primitive member as claimed, Bennion[7]),

Bennion does not explicitly teach said plurality of primitive data members in the

binary fragment payload represent "all" the primitive data members associated with said

object. However, since each record within the COMPANY object may be classified as

either a container record or a data-containing record (i.e. a primitive data member) (col.

3, lines 19-22, wherein the "Company" (Figure 6, item 603, Bennion) corresponds to the

object claimed, Bennion) and also since "COMPANY DATA 601" is a data-containing

record that contains data only (not other records) and also since it "describes the

company as a whole", then it is suggested that data-containing record "COMPANY

DATA 601" must represent all the primitive data members associated with the object

"COMPANY" (Col. 3, lines 5 – 8, Bennion). Therefore, it would have been obvious to

one having ordinary skill in the art at the time the invention was made to include all of

---

[7] Examiner makes note that the specification provides examples of "primitive members", (specification of
the disclosure; [0006], "string", "integer") also as defined by the dictionary ("Academic Press Dictionary of
Science and Technology from Elsevier Science & Technology, Copyright 1992, 1996 by Academic
Press") Primitive is a fundamental unit that cannot be divided. Therefore, data (in a data-containing
record which does NOT contain other records) is of a primitive data type. See also, figure 3, Bennion, first
block of data starting from the left on "151-200", the payload "DATA" includes "123 Avenue "E" " which
are string and integers. Examiner has interpreted the claims in light of the specification. See *In re Van
Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

the primitive data members of the object in a binary fragment payload as taught by

Bennion to provide a complete COMPANY object record as shown in the example given

in col. 3 lines 19-31. One of ordinary skill in the art would be motivated to make the

aforementioned combination with reasonable expectation of success.

Bennion also discloses:

wherein said plurality primitive data members are in a storage engine record

format (col. 3, lines 19-24, "Thus, a hierarchical structure emerges. Each individual

record contained within COMPANY record 600 may be classified as either a container

record or a data-containing record. Container records contain other records, while data-

containing records contain data. No record is both a container record and a data-

containing record"; col. 4, lines 40-43, "By definition, a data-containing record has a

"Length" equal to the actual length of the record itself, since a data-containing record

cannot contain another record", wherein a data-containing record is a primitive data

member, Bennion),

wherein the binary fragment header comprises a type field (col. 1, lines 56-60,

"The data structure defined by the present invention includes two types of records: data-

containing records and container records. Data-containing records contain data, while

container records contain other records. A code point found at the beginning of each

record specifies its type.", wherein the data-containing record type is equivalent to a

binary fragment type) and a length field (Figure 2; col. 1, lines 60-62, "A length field

facilitates variable data lengths for data-containing records", Bennion), and

   wherein the type field indicates that the binary fragment payload is devoid of any

non-primitive data members (Fig. 6, item 601, Col. 3, lines 19 – 27, "…Container

records contain other records, while data-containing records contain data. No record is

both container record and a data-containing record…" wherein the Examiner interprets

that since item 601 "COMPANY DATA", Fig. 6, does not contain other records, as

opposed to "REGION 602", then it is devoid of any non-primitive member as claimed,

Bennion[8]).

   Bennion does not explicitly disclose the claimed "type field indicates that the

binary fragment is the only fragment of the object".

   On the other hands, Roy discloses that the type field indicates that the binary

fragment is the only fragment of the object (col.3, lines 52-59, "The remaining sixty-eight

bits of the PDU are used for various other addressing information such as indicating

whether the PDU contains an ATM cell, a packet, or a control message, whether

reassembly of the packet should be aborted, whether the payload is a first fragment,

middle fragment or last fragment, how many payload bytes are in the last fragment, the

fragment sequence count, and a destination flow identifier", wherein the fragment

sequence count is used to show the fragment count, Roy). It would have been obvious

to one having ordinary skill in the art at the time the invention was made to incorporate

---

[8] Examiner makes note that the specification provides examples of "primitive members", (specification of
the disclosure; [0006], "string", "integer") also as defined by the dictionary ("Academic Press Dictionary of
Science and Technology from Elsevier Science & Technology, Copyright 1992, 1996 by Academic
Press") Primitive is a fundamental unit that cannot be divided. Therefore, data (in a data-containing
record which does NOT contain other records) is of a primitive data type. See also, figure 3, Bennion, first
block of data starting from the left on "151-200", the payload "DATA" includes "123 Avenue "E" " which
are string and integers. Examiner has interpreted the claims in light of the specification. See *In re Van
Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

the type field indicating that the binary fragment is the only fragment of the object as

disclosed by Roy into the binary fragment data representation as disclosed in Bennion

for part of addressing information (col. 3, lines 52-54). One of ordinary skill in the art

would be motivated to make the aforementioned combination with reasonable

expectation of success.

Response to Arguments

16.   Applicant argues that; "Bennion's code point 202 does not indicate that the data

format is devoid of any non-primitive data members".

        Examiner respectfully disagrees. First, in response to applicant's argument that

the references fail to show certain features of applicant's invention, it is noted that the

features upon which applicant relies (i.e., "data format is devoid...") are not recited in

the rejected claim(s). Although the claims are interpreted in light of the specification,

limitations from the specification are not read into the claims. See *In re Van Geuns*, 988

F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Second, Bennion does disclose: the

type field indicates that the binary fragment payload is devoid of any non-primitive data

members (Fig. 6, item 601, Col. 3, lines 19 – 27, "...Container records contain other

records, while data-containing records contain data. No record is both container record

and a data-containing record..." wherein the Examiner interprets that since item 601

"COMPANY DATA", Fig. 6, does not contain other records, as opposed to "REGION

602", then it is devoid of any non-primitive member as claimed, Bennion[9]).

17.   Applicant argues that; "Bennion's code point 202 does not indicate a property of

a collection that is in another record".

        Examiner respectfully disagrees. First, in response to applicant's argument that

the references fail to show certain features of applicant's invention, it is noted that the

features upon which applicant relies (i.e., "collection that is in another record…") are not

recited in the rejected claim(s). Although the claims are interpreted in light of the

specification, limitations from the specification are not read into the claims. See *In re*

Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Second, the

combination of Bennion, in view of Krishnaprasad, Sarkar, and Stickler does disclose

the claimed limitation: the bit field indicates a property of a collection in one of a plurality

of collection element fragments (Col. 7, lines 40 – 49, and Col. 17, lines 45 – 47,

Stickler)

---

[9] Examiner makes note that the specification provides examples of "primitive members", (specification of
the disclosure; [0006], "string", "integer") also as defined by the dictionary ("Academic Press Dictionary of
Science and Technology from Elsevier Science & Technology, Copyright 1992, 1996 by Academic
Press") Primitive is a fundamental unit that cannot be divided. Therefore, data (in a data-containing
record which does NOT contain other records) is of a primitive data type. See also, figure 3, Bennion, first
block of data starting from the left on "151-200", the payload "DATA" includes "123 Avenue "E" " which
are string and integers. Examiner has interpreted the claims in light of the specification. See *In re Van
Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Points of Contact

Any inquiry concerning this communication or earlier communications from the examiner should be directed to GIOVANNA COLAN whose telephone number is (571)272-2752. The examiner can normally be reached on 8:30 am - 5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Giovanna Colan
Examiner
Art Unit 2162
November 15, 2008

/John Breene/
Supervisory Patent Examiner, Art Unit 2162